

FIG. 1

Variable

w_i	= normalized weight of flow i ;
cw	= index of the current window being served;
fw_i	= index of the window containing flow i 's last packet;
c_i	= net credit for flow i ;
$pw[k]$	= pointer to Window k ;
$pw[k].nw$	= index of the next window from Window k ;
P_i	= newly arriving packet for flow i ;
$ P_i $	= size of packet P_i ;

Initialization() /* idle to busy */

1. $cw \leftarrow 1$;
2. **for** (each flow i) **do** $fw_i \leftarrow 1$; $c_i \leftarrow w_i$; **endfor**

Arrival(P_i)

 Compute the index of the window where P_i can be placed;

1. $m \leftarrow 0$;
 2. **while** ($c_i < |P_i|$) **do** $c_i \leftarrow w_i + c_i$; $m \leftarrow m + 1$; **endwhile**
 3. **if** (there is no flow- i packet in the queue) $m = \text{uniform}(0, m)$; **endif**
 4. $fw_i \leftarrow fw_i + m$;
- Place the packet in the window;
5. **if** ($pw[fw_i]$ does not exist) $\text{Create_Window}(pw[fw_i])$; **endif**
 6. $\text{Enqueue}(pw[fw_i], P_i)$;

 Update the credit;

7. $c_i \leftarrow c_i - |P_i|$;

Departure()

 Remove the packet from the head of the first window;

1. $\text{Dequeue}()$;
- Update variables and pointers if the window becomes empty;
2. **if** (the window is empty)
 3. **for** (each flow i such that $fw_i = cw$)
 4. $fw_i \leftarrow pw[cw].nw$; $c_i \leftarrow w_i$; **endfor**
 5. $cw \leftarrow pw[cw].nw$; **endif**

FIG. 2

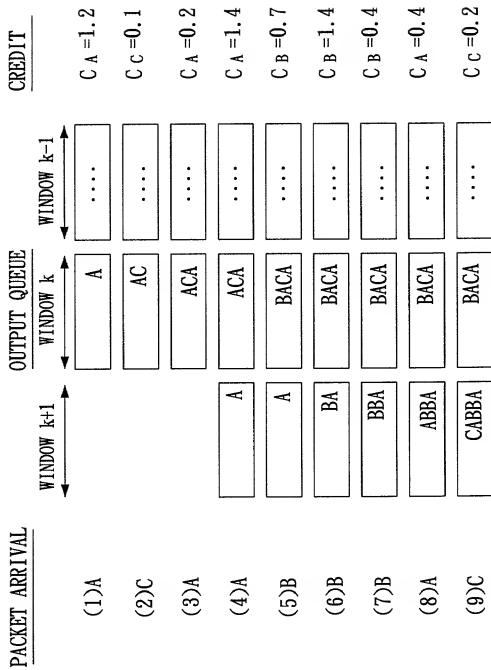


FIG. 3